

## **REMARKS**

Claims 1, 3-22, 24-43 and 45-53 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

### **Request for Supervisory Examiner Review Pursuant to M.P.E.P. § 707.02:**

Applicants note that this is the eighth different office action received in this application. Applicants also note that this application has been pending for over seven years. Such long pending applications which have received numerous actions are addressed by M.P.E.P. § 707.02. Pursuant to M.P.E.P. § 707.02, Applicants request that a Supervisory Patent Examiner review this application “with a view to finally concluding its prosecution.” According to M.P.E.P. § 707.02, this application “should be carefully studied by the supervisory patent examiner and every effort should be made to terminate its prosecution.” As noted in Applicants’ remarks below, the application is clearly in condition for allowance. **If there are any questions concerning the allowability of the present application, Applicants’ undersigned attorney earnestly requests a telephone conference with the Supervisory Patent Examiner to expeditiously resolve any outstanding issues.** Applicants also note that according to M.P.E.P. § 707.02, this application “is to be considered ‘special’ by the examiner.” Applicants also strongly object to the “piecemeal examination” that has been applied to this application. See M.P.E.P. § 707.07(g).

### **Section 103(a) Rejections:**

The Examiner rejected claims 1, 3-7, 9-22, 24-29, 31-40, 42, 43, 45 and 47-52 under 35 U.S.C. § 103(a) as being unpatentable over Merk et al. (U.S. Patent 6,754,886) (hereinafter “Merk”) in view of Carper (U.S. Patent 6,256,690), and claims 8, 30, 41, 46 and 53 as being unpatentable over Merk and Carper in view of Erickson et al. (U.S. Patent 6,851,089) (hereinafter “Erickson”). Applicants respectfully traverse these rejections for at least the following reasons.

Regarding claim 1, in contrast to the Examiner's contention, Merk in view of Carper fails to teach or suggest **generating a computer programming language object from a data representation language representation of the object**. The Examiner relies on FIG. 3, item 150 and column 6, lines 35-60 of Merk. However, the cited passage refers to reconstructing an object stored via Java's Object Serialization functionality, which as is well known in the art, uses a binary format to store object information (Merk, column 4, lines 12-21 and 45-49). Merk teaches storing, on a smartcard, a sequence of bytes representing the object and its current status by storing information about the object, including attributes and their names, as well as additional objects referenced by the first object. However, nowhere does Merk describe generating a computer programming language object *from a data representation language representation of the object*. Nor does Merk mention anything about a data representation language representation of objects. Merk's Java-based serialization store does not utilize a data representation language to store object information and therefore does not generate an object from a data representation language representation of the object.

Carper also fails to teach or suggest, even if combined with Merk, generating a computer programming language object from a data representation language representation of the object, and therefore fails to overcome Merk's failure to teach or suggest this limitation of Applicants' claim.

In further regards to claim 1, the Examiner's combination of Merk in view of Carper also fails to teach or suggest **a client device receiving a message in the data representation language from a service device in the distributed computing environment prior to generating the computer programming language object, wherein the message includes the data representation language representation of the object**. The Examiner admits that Merk fails to teach this limitation, but relies on Carper, citing column 7, lines 1-10, 35-45 and line 65 – column 8, line 11. Carper teaches a

smart card system that allows multiple applications to be executed simultaneously and in which applications may send messages to each other (Carper, column 4, 37-45).

Firstly, Carper, even if combined with Merk, does not teach or suggest **a client device receiving a message from a service device in the distributed computing environment**. Instead, Carper teaches that a dormant application is “initialized” when the application “first receives a command indicting the application” (Carper, column 7, lines 1-10). Thus, Carper teaches that one application is initialized (from dormancy) when it receives a command from another application *on the same device*. Carper does not mention, either at the Examiner’s cited passages or elsewhere, anything regarding a client device receiving a message *from a service device*, as recited in Applicants’ claim. Furthermore, Merk also fails to teach or suggest a client device receiving a message from a service device, as admitted by the Examiner. Thus, the Examiner’s combination of Merk and Carper fails to teach or suggest a client device receiving a message from a service device in the distributed computing environment.

Moreover, Merk in view of Carper fails to teach or suggest that **the message (received by a client device from a service device) includes a data representation language representation of the object**, as recited in Applicants’ claim. The Examiner relies on modifying Merk to include the message receiving taught by Carper. However, Carper’s messages do not include a data representation language representation of a computer programming language object. Instead, Carper teaches that his application receive a command, and in response, are initialized (or re-initialized) including loading any previously stored data object. However, Carper’s command messages do not include data representation language representations of objects.

Even if Carper were combined with Merk, the resulting system would not include a client device receiving a message from a service device that includes a data representation language representation of a computer programming language object. Instead, the Examiner’s suggested combination of cited art would result in a system in which dormant, smart card based, applications can be re-initialized upon receiving a

command message. The re-initialized application may then load stored objects as taught by either Merk or Carper.

No combination of Merk and Carper would include a message that includes a data representation language representation of a computer programming language object since neither Merk nor Carper, whether considered singly or in combination, teaches or suggests such a message, as recited in Applicants' claim.

To establish a *prima facie* obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. As shown above, the Examiner's combination of cited art fails to teach or suggest **generating a computer programming language object from a data representation language representation of the object** and also fails to teach or suggest **a client device receiving a message in the data representation language from a service device in the distributed computing environment prior to generating the computer programming language object, wherein the message includes the data representation language representation of the object.** Thus the Examiner has failed to provide a *prima facie* rejection.

**Additionally, the Examiner has not provided a valid reason for combining Carper's message receiving capability in Merk's system.** The Examiner states that it would have been obvious to "modify the invention of Merk to include the features of Carper because this [would] clean-up memory space before the session ends" (Office Action dated July 13, 2007, page 4, lines 6-10). However, including the ability for a dormant application to be (re)initialized in response to receiving a command message, as taught by Carper (relied on by the Examiner) has nothing to do with cleaning up memory before a session ends. Thus, the Examiner's reason to modify the references is not commensurate with the modification proposed by the Examiner. Furthermore, the Examiner's stated reason does not make sense. One skilled in the art would not modify Merk to include (re)initializing dormant application that receive command messages in order to clean-up memory before a session ends. In contrast to the Examiner's statement,

(re)initializing a dormant application would not clean up any memory, but instead would use additional memory.

The rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 22 and 43.

Regarding claim 10, Merk in view of Carper **fails to teach or suggest the client device receiving a message in a data representation language from a service device in the distributed computing environment, wherein the message includes a data representation language representation of a computer programming language object.** Merk in view of Carper also **fails to teach or suggest generating the object from the data representation language representation of the object,** as recited in Applicants' claim. As shown above, regarding the rejection of claim 1, the Examiner's combination of Merk in view of Carper does not teach or suggest a client receiving a message *from a service device*, where the message is in a data representation language and includes a data representation language representation of a computer programming language object. Nor does a combination of Merk and Carper, as also shown above, teach or suggest generating the object from a data representation language representation of the object. Please refer to Applicants' remarks above regarding why Merk in view of Carper fail to teach this limitation.

Additionally, Merk in view of Carper fails to **teach or suggest determining if the user has access rights to the computer programming language object.** The examiner relies on column 8, lines 42-59 and column 10, lines 36-45 of Carper. However, Carper, even if combined with Merk does not teach or suggest determining if the user *has access rights to the computer programming language object*. Instead, Carper, at the Examiner's cited passages, teaches that applications may includes a "digital signature or similar authentication device" to "verify the authenticity of the message data" (emphasis added, Carper, column 10, lines 37-44). Additionally, Carper teaches (column 8, lines 44-58) a security algorithm "to verify the authenticity of terminal commands, user inputs, etc.). Thus, Carper teaches security measures in order that "data may be securely exchanged

between applications” (Carper, column 10, lines 43-44). However, Carpers’ verifying of messages between applications and authentication of terminal commands is not the same as, nor does it teach or suggest, even if combined with Merk, determining *if a user has access rights to a computer programming language object*, as recited in Applicants’ claim. Verifying that message is actually from another application or that a terminal command is valid does not have anything to do with determining whether a user has access rights to a computer programming language object.

Moreover, Merk in view of Carper does not teach or suggest **generating the object from the data representation language representation of the object if the user has access rights to the computer programming language object**. As shown above, the Examiner’s combination of Merk in view of Carper does not include determining if a user has access rights to a computer programming language object. Merk teaches storing and retrieving serialized data objects, but does not mention loading a serialized object if it is determined that a user has access rights to a particular object. Carper, even if combined with Merk, teaches (re)initializing a dormant application when that application receives a message or command, but also does not mention generating an object if it is determined that the user *has access rights* to the object.

The rejection of claim 10 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 32 and 47.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

## CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-47300/RCK.

Respectfully submitted,

/Robert C. Kowert/  
Robert C. Kowert, Reg. #39,255  
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: October 15, 2007